Module Code & Module Title

CS4001NI Programing

Assessment Weightage & Type

30% Individual Coursework

2022-23 Autumn

Student Name: Sonam Dhendup Gurung

London Met ID: 22067100

College ID: NP01NT4A220055

Assignment Due Date: 10 May 2023

Assignment Submission Date: 10 May 2023

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of contents

Contents

Table of figures

Table of tables:

Introduction

Introduction to the project

The goal of this project is to use Java's object-oriented idea to implement a real-world problem situation. To that end, a class representing a bank card GUI and its two subclasses, which stand in for a debit card GUI and a credit card GUI, respectively, will be created.

Aim:

The purpose of this task is to add a class to the project you created for the first section of the coursework to create a graphical user interface (GUI) for a system that maintains information about bank cards in an Array List. The class's primary method will be tested via the command prompt, and it will be included within the class.

Tools used in the following program:

1.Blue J

2.Diagramly

3.Microsoft word

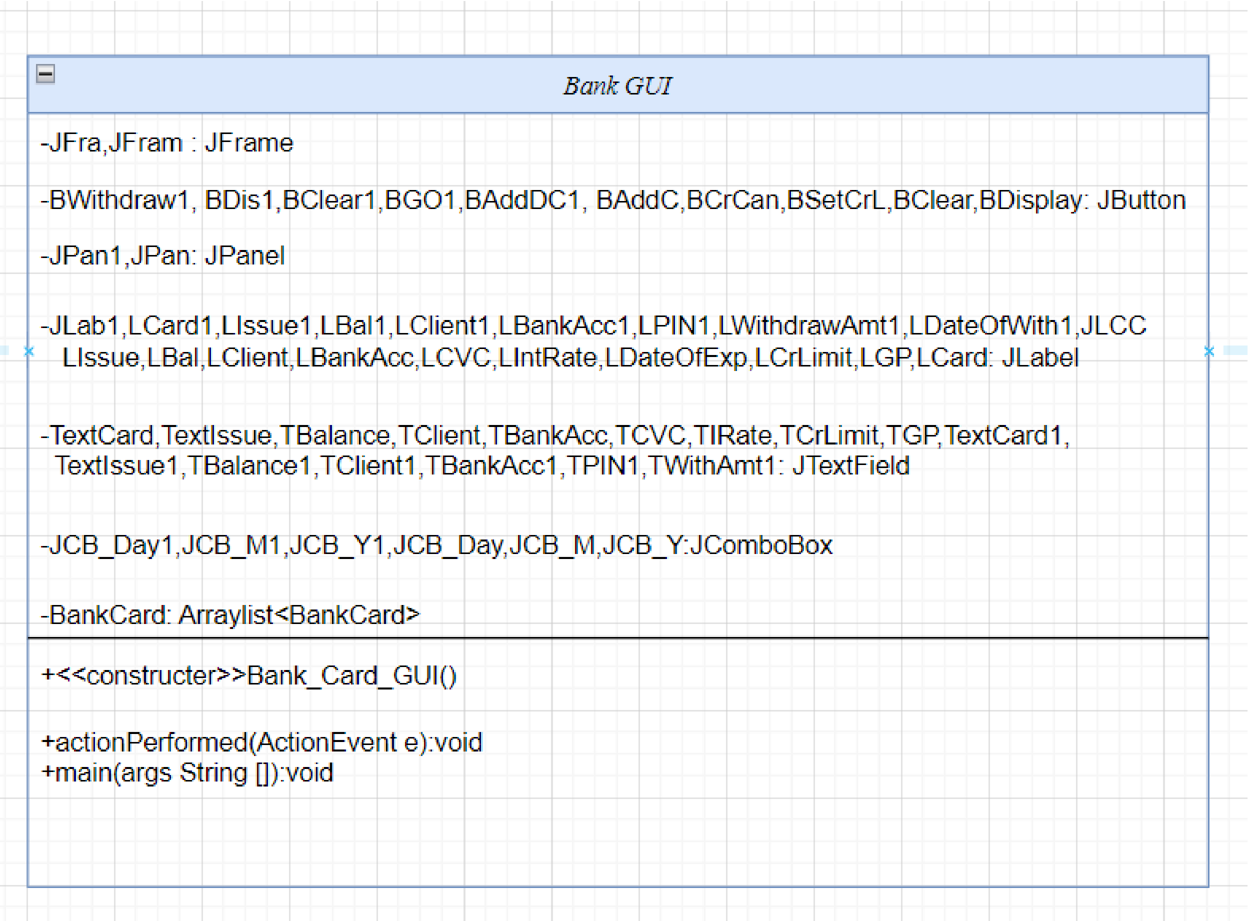
1.Blue J: A Java integrated development environment called BlueJ was created primarily with educational objectives in mind (Ali, 2018). With the aid of the JDK (Java Development Kit), it functions. Blue was an integrated system with its own environment and programming language (Ali, 2018). BlueJ is the Java programming language's implementation of the blue environment design (Ali, 2018).

Figure 1:Blue J

2.Diagramly: Diagramly is a free online tool for creating diagrams and charts. (Gibson, 2011)

3.Microsoft word: Microsoft Word is a word processing application that enables the generation of both straightforward and intricate texts (Anon., n.d.). You will have access to both the online and download versions of the application with Office 365. With the online version, you may instantly share and work on documents with other people (Anon., n.d.).

Class Diagram



Bank GUI

-JFra,JFram : JFrame

-BWithdraw1, BDis1,BClear1,BGO1,BAddDC1, BAddC,BCrCan,BSetCrL,BClear,BDisplay: JButton

-JPan1,JPan: JPanel

-JLab1,LCard1,LIssue1,LBal1,LClient1,LBankAcc1,LPIN1,LWithdrawAmt1,LDateOfWith1,JLCC LIssue,LBal,LClient,LBankAcc,LCVC,LIntRate,LDateOfExp,LCrLimit,LGP,LCard: JLabel

-TextCard,TextIssue,TBalance,TClient,TBankAcc,TCVC,TIRate,TCrLimit,TGP,TextCard1, TextIssue1,TBalance1,TClient1,TBankAcc1,TPIN1,TWithAmt1: JTextField

-JCB_Day1,JCB_M1,JCB_Y1,JCB_Day,JCB_M,JCB_Y:JComboBox

-BankCard: Arraylist<BankCard>

+<<constructer>>Bank_Card_GUI()

+actionPerformed(ActionEvent e):void
+main(args String []):void

Pseudocode:

Debit Card GUI

```
Import JPanel from javax.swing package,

Import JLabel from java.awt package,

Import JTextField from javax.swing package,

Import Font from java.awt package,

Import Color from javax.swing package,

Import JRadioButton from javax.swing package,

Import ButtonGroup from javax.swing package,

Import JFrame from javax.swing package,

Import JComboBox from javax.swing package,

Import JButton from javax.swing package,

Import JCheckBox from javax.swing package,

Import ActionListener from java.awt.event package,

Import ActionEvent from java.awt.event package,

Import ArrayList from java.util package,

Import JOptionPane from javax.swing package

CREATE a class Bank_Card_GUI which implements ActionListener interface.

DO

CREATE a

For Debit

DECLARE an instance variable JButton BWithdraw1, BDis1, BClear1, BG01,BAddDC having access modifier private

DECLARE an instance variable JPanel JPan1 having access modifier private.
DECLARE an instance variable JLabel JLab1, LCard1, LIssue1, LBal1, LClient1, LBankAcc1, LPIN1, LWithdrawAmt1, LDateOfWith1 having access modifier private

DECLARE an instance variable JComboBox JCB_Day1, JCB_M1, JCB_Y1 having access modifier private.

DECLARE an instance variable JTextField TextCard1, TextIssue1, TBalance1, TClient1, TBankAcc1, TPIN1, TWithAmt1 having access modifier private.

DECLARE an instance variable JFrame JFra1 having access modifier private.

For Credit

DECLARE an instance variable JButton BSetCrl, BDisplay,BClear,BCrCan,BAddC having access modifier private

DECLARE an instance variable JLabel JLCC, LCard, LIssue, LBal, LClient, LBankAcc , LCVC, LintRate , LDateOfExp, LCrLimit, LGP having access modifier private

DECLARE an instance variable JComboBox JCB_Day, JCB_M, JCB_Y having access modifier private.

DECLARE an instance variable JTextField TextCard, TextIssue, TBalance, TClient, TBankAcc, TCVC, TIRate,TCrLimit,TGP having access modifier private

DECLARE an instance variable JFrame JFram having access modifier private.

DECLARE an instance variable JPanel JPan having access modifier private.

CREATE an empty ArrayList of type BankCard called Arr_List

DECLARE a function actionPerformed that takes an ActionEvent object as an argument.

DO

 If the e.getSource() is equal to BWithdraw1:

DO

Convert the text value of WithAmt1 to an integer and assign it to Withdraw_Amount1.
```

Convert the text value of TextCard1 to an integer and assign it to Card_id1.

Convert the text value of TPIN1 to an integer and assign it to Pin_number1.

Get the day, month, and year values from the corresponding JComboBoxes, and combine them into a Dr_date_value1.

Show a message dialog indicating that the withdrawal was successful.

DO

If the (RR)object is an instance of DebitCard:

DO

Downcast the BankCard object to a DebitCard object

Compare the card ID of the DebitCard object to the input card_id1

DO

If they match, show a message dialog indicating the card ID, withdrawal amount, and PIN number.

If the input Pin_number1 matches the PIN number of the DebitCard object

DO

If the Withdraw_amount is less than or equal to the balance amount of the DebitCard object

DO

Call the withdraw method of the DebitCard object with the Withdraw_amount1, Dr_date_value1, and Pin_number1 as arguments.

Else show a message dialog indicating that there is insufficient amount.

Else show a message dialog indicating that the PIN number is incorrect.

Else show a message dialog indicating that the card ID doesn't match.

```
END
END
END
END
IF the event source is BDis1
DO
FOR each Bankcard arr1 in Arr_List DO
IF arr1 is an instance of Debitcard THEN
SET obj as Debitcard instance from arr1
CALL obj.display() method
END IF
END FOR
END IF
ELSE IF the event source is equal to BClear1.
DO
Set the text fields TextCard1, TextIssue1, TBalance1, TClient1, TBankAcc1, and TPIN1 to an empty string.
END
END IF the event source is BGO1.
DO
mC();
JFram.dispose();
END
END IF the event source is BAddDC1,
DO
Take the integer value of TextCard1 and assign it to Card_id.
Take the string value of TextIssue1 and assign it to IssuerBankk.
Take the integer value of TBalance1 and assign it to Balance_Amt.
Take the string value of TClient1 and assign it to Client_Name.
Take the string value of TBankAcc1 and assign it to Bank_Amount.
Take the integer value of TPIN1 and assign it to PIN_Number.
INITIALIZE the boolean variable B1 to true.
For each BankCard object BankC1 in the Arr_List:
DO
If BankC1 is an instance of DebitCard:
Cast BankC1 to DebitCard and assign it to DEBC.
If the card ID of DEBC equals Card_id:
DO
    Set B1 to false.
    Break out of the loop.
END
END
END
If B1 is true, then:
DO
CREATE a new DebitCard object obj_Deb with the parameters Balance_Amt, Card_id, Bank_Amount, IssuerBankk, Client_Name, and PIN_Number.
```

```
Add obj_Deb to the Arr_List ArrayList.
Show a message dialog box with the message "Successfully added!".
END
Else
DO
show a message dialog box with the message "Already Added".
For Credit Card
If the e.getSource() is equal to BAddC1:
DO
Convert the text value of TextCard to an integer and store it in LCard.
Get the text value of TextIssue and store it in LIssue.
Convert the text value of TBalance to an integer and store it in LBal.
Get the text value of TClient and store it in LClient.
Get the text value of TBankAcc and store it in LBankAcc.
Get the text value of TGP and store it in LGP.
Convert the text value of TCrLimit to an integer and store it in LCrLimit.
Convert the text value of TCVC to an integer and store it in LCVC.
Convert the text value of TIRate to an integer and store it in LIntRate.
Set the boolean variable B1 to true.
Loop through each BankCard object (BankC) in the ArrayList named Arr_List.
DO
Check if the current BankCard object (BankC) is an instance of CreditCard.
DO
```

```
If it is an instance of CreditCard, cast the BankCard object to CreditCard and get its CID.
If the CID of the CreditCard object matches LCard,
DO
set B1 to false and break out of the loop.
END
END
END
If B1 is still true,
DO
CREATE a new CreditCard object named obj_Cr with the parameters LBal, LCard, LBankAcc, LIssue, LClient, LCVC, LIntRate, and LGP.
Add the new CreditCard object(obj_Cr) to the Arr_List ArrayList.
Display a success message using JOptionPane if the CreditCard object was successfully added.
END
ELSE
DO
Display an error message using JOptionPane if the CreditCard object was already added.
END
END
ELSE IF get event Source () is equal to BCrCan
DO
```

```
Try
DO
Get the card ID entered in the TextCard field and convert it to an integer
Set boolean id to false
Loop through each BankCard B_Card in Arr_List
DO
IF B_Card is an instance of CreditCard and its CID matches the entered Card_Id then.
DO
Set id to true
CREATE a new CreditCard instance C_Card from B_Card
Call C_Card.CancelCreditCard() method to cancel the credit card
Show a success message using JOptionPane.showMessageDialog()
If id is still false
DO
 then show a warning message using JOptionPane.showMessageDialog() indicating that the entered Card_Id doesn't exist and ask to enter a correct Card_Id
END
CATCH (Exception e)
DO
If an exception occurs during the execution of try block, then show an error message using JOptionPane.showMessageDialog() indicating that the credit card details are invalid.
END DO
```

END TRY.

END DO

ELSE IF get event Source () is equal to BSetCrL

DO

Get the text value of TCrLimit in string and store it in CR_Limit.

Convert the text value of TextCard to an integer and store it in Card_id.

Convert the text value of TCVC to an integer and store it inCVC_Number.

Get the day, month, and year values from the corresponding JComboBoxes, and combine them into a Cr_date_value.

Show a message dialog indicating that the withdrawal was successful.

Check if UD is an instance of DebitCard

If UD is a DebitCard, then:

Downcast UD to a CreditCard instance Cr_obj

Check if the Card_id entered by the user matches the CID of Cr_obj

If the Card_id matches, then:

Show a message using JOptionPane.showMessageDialog() to display the card ID, credit limit and CVC number.

Check if the CVC_Number entered by the user matches the cvc_Number of Cr_obj

If the CVC_Number matches, then:

Check if the withdrawal amount is less than or equal to the balance amount in Cr_obj

If the withdrawal amount is less than or equal to the balance amount, then:

Set the new credit limit and grace period using Cr_obj.setCredit_Limit()

Show a success message using JOptionPane.showMessageDialog()

```
If the withdrawal amount is greater than the balance amount, show an error message using JOptionPane.showMessageDialog() indicating insufficient amount

If the CVC_Number entered by the user does not match the cvc_Number of Cr_obj, show an error message using JOptionPane.showMessageDialog() indicating incorrect pin number

If the Card_id entered by the user does not match the CID of Cr_obj, show an error message using JOptionPane.showMessageDialog() indicating incorrect card ID

END

ELSE IF get event source() is equals to BClear

DO

Set the text fields TextCard, TextIssue, TBalance, TClient, TBankAcc, and TCVC to an empty string.

END

ELSE IF get event source () is equals to BDisplay

DO

For each BankCard arr in Arr_List, do the following:

DO

IF arr is an instance of CreditCard

DO

IF arr is a CreditCard, then:

Downcast arr to a CreditCard instance obj

Call the Display_All_Information() method of obj to display all the information of the credit card

END IF
END DO
END DO
END
CREATE a instance class main1D()
DO
CREATE a new JFrame object called JFra1 and set the title of JFra1 to "Coursework.”
Set the position of JFrame JFra1 to (250-xaxis, 5-yaxis) and to (1000-width, 800-height)
Set layout to null.
CREATE a new JPanel object called JPan1.
Set the position of JPan1 to (0-xaxis, 0-yaxis) and to (1000-width, 800-height)
Set background COLOR. lightGray
Set layout to null.
CREATE a new JLabel object called JLab1 and set the title of JLab1 to "Debit Card.”
Set the position of JLab1 to (400-xaxis, 20-yaxis) and to (200-width, 60-height)
Set Font to Arial, BOLD, size 30
CREATE a new JLabel object called LCard1 and set the title of LCard1 to "Card ID.”
Set the position of LCard1 to (80-xaxis, 80-yaxis) and to (80-width, 50-height)
Set Font to Arial, PLAIN, size 20.
CREATE a new JTextField object called TextCard1.
Set the position of TextCard1 to (160-xaxis, 90-yaxis) and to (100-width, 30-height)
CREATE a new JLabel object called LIssue1 and set the title of LIssue1 to "Issuer Bank.”
Set the position of LIssue1 to (600-xaxis, 80-yaxis) and to (140-width, 50-height)
Set Font to Arial, PLAIN, size 20.
```

CREATE a new JTextField object called TextIssue1.

Set the position of TextIssue1 to (725-xaxis, 90-yaxis) and to (100-width, 30-height)

CREATE a new JLabel object called LBal1 and set the title of LBal1 to "Balance Amount."

Set the position of LBal1 to (80-xaxis, 150-yaxis) and to (200-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TBalance1.

Set the position of TBalance1 to (240-xaxis, 150-yaxis) and to (100-width, 30-height)

CREATE a new JLabel object called LClient1 and set the title of LClient1 to "Client Name."

Set the position of LClient1 to (600-xaxis, 150-yaxis) and to (150-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TClient1.

Set the position of TBalance1 to (725-xaxis, 150-yaxis) and to (100-width, 30-height)

CREATE a new JLabel object called L BankAcc1 and set the title of L BankAcc1 to "Bank Account."

Set the position of LBankAcc1 to (80-xaxis, 220-yaxis) and to (200-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TBankAcc1.

Set the position of TBankAcc1 to (220-xaxis, 220-yaxis) and to (120-width, 30-height)

CREATE a new JLabel object called LPIN1 and set the title of LPIN1 to "Pin Number."

Set the position of LPIN1 to (600-xaxis, 220-yaxis) and to (150-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TPIN1.

Set the position of TPIN1 to (725-xaxis, 220-yaxis) and to (100-width, 30-height)

CREATE a JButton object called BAddDC1 and set its title to “Add Debit Card.”

Set the position of BAddDC1 to (350-xaxis, 300-yaxis) and to (190-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a JButton object called BGO1 and set its title to “Go Back.”

Set the position of BGO1 to (650-xaxis, 350-yaxis) and to (150-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JLabel object called LWithdrawAmt1.

Set the position of LWithdrawAmt1 to (80-xaxis, 425-yaxis) and to (200-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TWithAmt1 and set its title “Withdraw Amount.”

Set the position of TWithAmt1 to (260-xaxis, 425-yaxis) and to (100-width, 30-height)

CREATE a new JLabel object called LDateOfWith1 and set its title “Date of Withdrawal.”

Set the position of LDateOfWith1 to (80-xaxis, 550-yaxis) and to (200-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a JComboBox and assign it to JCB_Day1.

Set the bounds of JCB_Day1 to (280-xaxis, 555-yaxis, 50-width, 20-height).

Use a for loop to add integers from 1 to 30 to JCB_Day1.

CREATE a String array with the months of the year and assign it to the variable 'months'.

CREATE a JComboBox and assign it to JCB_M1. Set the bounds of JCB_M1 to (350-xaxis, 555-yaxis, 80-width, 20-height). Add the months array to JCB_M1.

CREATE another JComboBox and assign it to JCB_Y1. Set the bounds of JCB_Y1 to (480x-axis, 555y-axis, 60-width, 20-height).

Use a for loop to add integers from 1900 to 2023 to JCB_Y1.

CREATE a JButton with text "Withdraw" and assign it to BWithdraw1.

Set the bounds of BWithdraw1 to (280x-axis, 655y-axis, 120-width, 20-height)

Set its font to Arial, plain style, size 20.

CREATE a JButton with text "Display" and assign it to BDis1.

Set the bounds of BDis1 to (450x-axis, 655y-axis, 100-width, 20-height) and set its font to Arial, plain style, size 20.

CREATE another JButton with text "Clear" and assign it to BClear1.

Set the bounds of BClear1 to (620x-axis, 655y-axis, 80-width, 20-height) and set its font to Arial, plain style, size 20.

Add action listeners to the buttons BClear1, BDis1, BWithdraw1, BGO1, and BAddDC1.

Add JCB_Y1, JCB_M1, JCB_Day1, LDateOfWith1, TWithAmt1, LWithdrawAmt1, BWithdraw1, BDis1, BClear1, BGO1, BAddDC1, LPIN1, TPIN1, TBankAcc1, LBankAcc1, TClient1, LClient1, TBalance1, LBal1, TextIssue1, LIssue1, TextCard1, LCard1and JLab1 to the JPanel JPan1.

Add JPan1 to JFra1 of JFrame.

Set JFra1 to visible.

For Credit Card

CREATE a instance class mC()

DO

```
CREATE a new JFrame object called JFram and set the title of JFram to "Coursework.”
Set the position of JFram to (250-xaxis, 5-yaxis) and to (1000-width, 800-height)
Set layout to null.
CREATE a new JPanel object called JPan.
Set the position of JPan to (0-xaxis, 0-yaxis) and to (1000-width, 800-height)
Set background COLOR.GRAY
Set layout to null.
CREATE a new JLabel object called JLCC and set the title of JLCC to "Credit Card.”
Set the position of JLCC to (400-xaxis, 20-yaxis) and to (200-width, 60-height)
Set Font to Arial, BOLD size 30.
CREATE a new JLabel object called LCard and set the title “Card ID”.
Set the position of LCard to (80-xaxis, 80-yaxis) and to (80-width, 50-height)
Set Font to Arial, PLAIN size 20.
CREATE a new JTextField object called TextCard.
Set the position of TextCard to (160-xaxis, 90-yaxis) and to (100-width, 30-height)
CREATE a new JLabel object called LIssue and set the title “Issuer Bank”.
Set the position of LIssue to (600-xaxis, 80-yaxis) and to (140-width, 50-height)
Set Font to Arial, PLAIN size 20.
CREATE a new JTextField object called TextIssue.
Set the position of TextIssue to (725-xaxis, 90-yaxis) and to (100-width, 30-height)
CREATE a new JLabel object called LBal and set the title “Balance Amount”.
Set the position of LBal to (80-xaxis, 150-yaxis) and to (200-width, 30-height)
Set Font to Arial, PLAIN size 20.
```

CREATE a new JTextField object called TBalance.

Set the position of Tbalance to (240-xaxis, 150-yaxis) and to (100-width, 30-height)

CREATE a new JLabel object called LClient and set the title “Client Name”.

Set the position of LClient to (600-xaxis, 150-yaxis) and to (150-width, 30-height)

Set Font to Arial, PLAIN size 20.

CREATE a new JTextField object called TClient.

Set the position of TClient to (725-xaxis, 150-yaxis) and to (100-width, 30-height)

CREATE a new JLabel object called LBankAcc and set the title of LBankAcc to "Bank Account.”

Set the position of LBankAcc1 to (80-xaxis, 220-yaxis) and to (200-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TBankAcc.

Set the position of TBankAcc to (220-xaxis, 220-yaxis) and to (120-width, 30-height)

CREATE a new JLabel object called LCVC and set the title of LCVC to "CVC Number.”

Set the position of LCVC to (600-xaxis, 220-yaxis) and to (150-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TCVC.

Set the position of TCVC to (725-xaxis, 220-yaxis) and to (100-width, 30-height)

CREATE a JButton object called BAddC and set its title to “Add Credit Card.”

Set the position of BAddC to (350-xaxis, 300-yaxis) and to (190-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a JButton object called BCrCan and set its title to “Cancel Credit.”

Set the position of BCrCan to (650-xaxis, 300-yaxis) and to (160-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JLabel object called LIntRate and set its title to “Interest Rate.”

Set the position of LIntRate to (80-xaxis, 350-yaxis) and to (150-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TIRate.

Set the position of TIRate to (220-xaxis, 350-yaxis) and to (150-width, 30-height)

CREATE a new JLabel object called LDateOfExp and set its title to “Expiration Date.”

Set the position of LDateOfExp to (80-xaxis, 425-yaxis) and to (200-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object called TIRate.

Set the position of TIRate to (220-xaxis, 350-yaxis) and to (150-width, 30-height)

CREATE a JComboBox and assign it to JCB_Day.

Set the bounds of JCB_Day1 to (280-xaxis, 430-yaxis, 50-width, 20-height).

Use a for loop to add integers from 1 to 30 to JCB_Day1.

CREATE a String array with the months of the year and assign it to the variable 'months'.

CREATE a JComboBox and assign it to JCB_M.

Set the bounds of JCB_M to (350-xaxis, 430-yaxis, 80-width, 20-height). Add the months array to JCB_M.

CREATE a JComboBox and assign it to JCB_Y.

Set the bounds of JCB_Y to (480x-axis, 430y-axis, 60-width, 20-height).

Use a for loop to add integers from 1900 to 2023 to JCB_Y.

CREATE a new JLabel object LCrLimit and set its title “Credit Limit”.

Set the bounds of LCrLimit to (80x-axis, 505y-axis, 150-width, 30-height)

CREATE a new JTextField object TCrLimit .

Set the bounds of TCrLimit to (250x-axis, 505y-axis, 150-width, 30-height).

CREATE a new JLabel object LGP and set its title “Grace Period”.

Set the bounds of LGPto (80x-axis, 580y-axis, 150-width, 30-height)

Set Font to Arial, PLAIN, size 20.

CREATE a new JTextField object TGP.

Set the bounds of TGP to (250x-axis, 580y-axis, 150-width, 30-height).

CREATE a JButton object called BSetCrl and set its title to “Set Credit Limit.”

Set the position of BSetCrl to (280-xaxis, 655-yaxis) and to (120-width, 20-height)

Set Font to Arial, PLAIN, size 20.

CREATE a JButton object called BDis and set its title to “Display.”

Set the position of BDis to (450-xaxis, 655-yaxis) and size to (100-width, 20-height)

Set Font to Arial, PLAIN, size 20.

CREATE a JButton object called BClear and set its title to “Clear.”

Set the position of BClear to (620-xaxis, 655-yaxis) and size to (80-width, 20-height)

Set Font to Arial, PLAIN, size 20.

Add action listeners to the buttons BClear, BDis, BSetCrl, BCrCan, and BAddC.

```
Add JCB_Y, JCB_M, JCB_Day, LDateOfExp, TIRate, TGP, BAddC, BDis, BClear, BCrCan, BSetCrl, LCVC, TCVC, TBankAcc, LBankAcc, TClient, LClient, TBalance, LBal, TextIssue, LIssue, TextCard, LCard and LGP, JLCC , LIntRate to the JPanel JPan.

Add JPan to JFram of JFrame.

Set JFram to visible.

END

END
```

Method Description:

Debit Card Button

Debit Card Button: Variable name of Debit Card Button- BAddDC1	BAddDC1 buttons when interacted: 1.Retrieves values entered in textfields such as TextCard1, TBankAcc1, TBalance1, TClient1, Tissue1 and TPIN1. 2.It checks whether a given Card_id already exists in the Arr_List of BankCard objects. 3.If the Card_id does not exist, it creates a new DebitCard object using the given parameters and adds it to the Arr_List. 4.If the Card_id already exists, it displays a message indicating that the card has already been added. 5. It shows a dialog “Successfully added” after adding the values in the text fields. If the same values are added again with matching details, it delievers a dialog “Already Added.”

2. Credit Card Button

Credit Card Button: Variable name of Credit Card Button- BAddC	BAddC buttons when interacted: 1. This Button firstly checks if the source of the event is the "Add Credit Card" button. If the button was indeed clicked, then it retrieves the values entered by the user from several text fields (TextCard, TextIssue, TBalance, TClient, TBankAcc, TGP, TCrLimit, TCVC, and TIRate) corresponding to different attributes of a Credit Card, such as its Card ID, balance, issuer bank, client name, credit limit, CVC, interest rate, etc. Next, the code checks if a Credit Card with the same Card ID already exists in the list of bank cards (Arr_List) by iterating through each object in that list. If such a Credit Card already exists, then the B variable is set to false, indicating that the Card ID is already taken. If it doesn't exist, then B remains true. Finally, if B is true, then the code creates a new CreditCard object with the values entered by the user and adds it to the Arr_List. It then displays a success message using the JOptionPane class. If B is false, then it simply displays an error message informing the user that the Card ID is already taken.

3.Withdraw Button:

Withdraw Button: Variable name of Withdraw Card Button-BWithdraw1	When we interact with the BWithdraw1 button, it extracts the values entered by the user for withdrawal amount, card ID, PIN number, and date of withdrawal. Then, it loops through the Arr_List, which is a list of objects of type BankCard. For each object, it checks if it is an instance of DebitCard. If it is, it downcasts it to DebitCard and compares its card ID with the user input card ID. If it matches, it displays the card information and checks if the user input PIN number matches the card's PIN number. If it matches, it checks if the withdrawal amount is less than or equal to the available balance of the card. If it is, it calls the withdraw() method of the DebitCard object to deduct the amount from the balance and update the transaction history. If the withdrawal amount is greater than the available balance, it shows an error message. If the user input PIN number does not match the card's PIN number, it shows an error message. If the user input card ID does not match the card's ID, it shows an error message. Finally, it displays a success message after the withdrawal is completed.

4.Credit Cancel Button

Cancel Credit Button: Variable name of Withdraw Card Button- BCrCan	When interacted with this button, firstly, it retrieves the value of a credit card ID entered in a text field. It then sets a boolean variable id to false, which will later be used to check whether the credit card with the given ID exists or not. Next, the code iterates through the Arr_List, which is a list containing objects of BankCard type. It checks if the object is an instance of CreditCard and if its CID matches the Card_id. If it finds a matching credit card, it sets the id variable to true, downcasts the BankCard object to a CreditCard object, and calls the CancelCreditCard() method on it to cancel the credit card. If the id variable is still false after the iteration, it means no credit card with the given ID was found, and it shows a warning message dialog with an appropriate message. If an exception is thrown during the execution of the code, it shows an error message dialog indicating that the credit card details are invalid.

5.Credit Limit Button

Set Credit Limit Button: Variable name of Credit Limit Button-SetCrL	When interacted with Set credit limit button, It retrieves input values such as the credit limit, card ID, CVC number, grace period, and card expiration date. It then iterates through an array list of bank cards to find a credit card matching the specified card ID. If a matching credit card is found, the code proceeds to validate the CVC number and the available balance on the account. If the CVC number is correct and the available balance is sufficient, the credit limit is updated with the new value specified by the user. If any errors occur during this process, such as an incorrect CVC number or insufficient funds, error messages are displayed to the user via pop-up dialogs.

6. Display Button

Display Button: Variable name Of Display Button: BDis1	BDis1 buttons displays all the information regarding the Debit Cards. It displays the values written in the balance amount , Card id , issuer bank , Bank Account , Withdrawal amount , Date of Withdrawal and Pin number.

7.Clear Button

Clear Button: Variable name of Clear Button- BClear1	The Clear button resets or clear all the values in the following buttons TextCard1, TextIssue1, TBalance1, TClient1,TBankAcc1 TPIN1, TWithAmt1

8.Go Back Button

Go Back Button: Variable name of Go Back Button- BGO1	The Go back Button takes you to the Credit Card GUI.

Test 1:

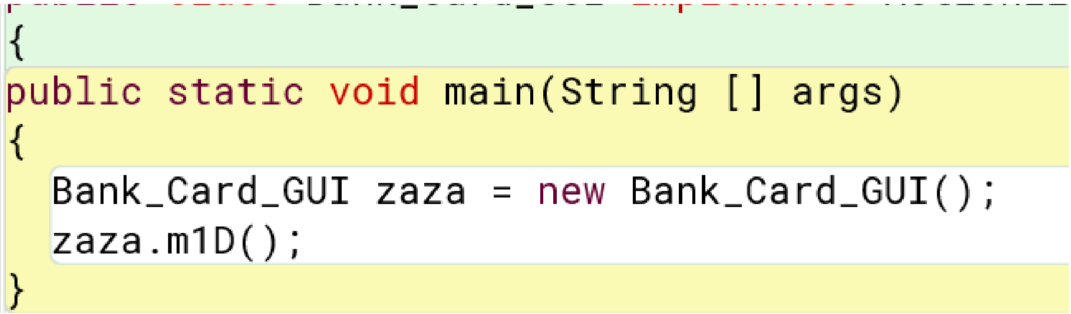
In this Test, the program is being run and compiled through command prompt. Here is the proof with screenshot.

Table 1: Run and Compile the program through command prompt

Test Number	1
Objective:	To test that the program can be compiled and run using the command prompt. The evidence should be shown through a screenshot.
Action:	Compile and run the program using command prompt. The evidence is shown through the screenshots.
Expected Result:	A method should be made in the class, the method is then used to compile and run the program through command prompt.
Actual Result:	A method was made in the class, the method was then used to compile and run the program through command prompt.
Conclusion	The Test is successful.

Evidence:

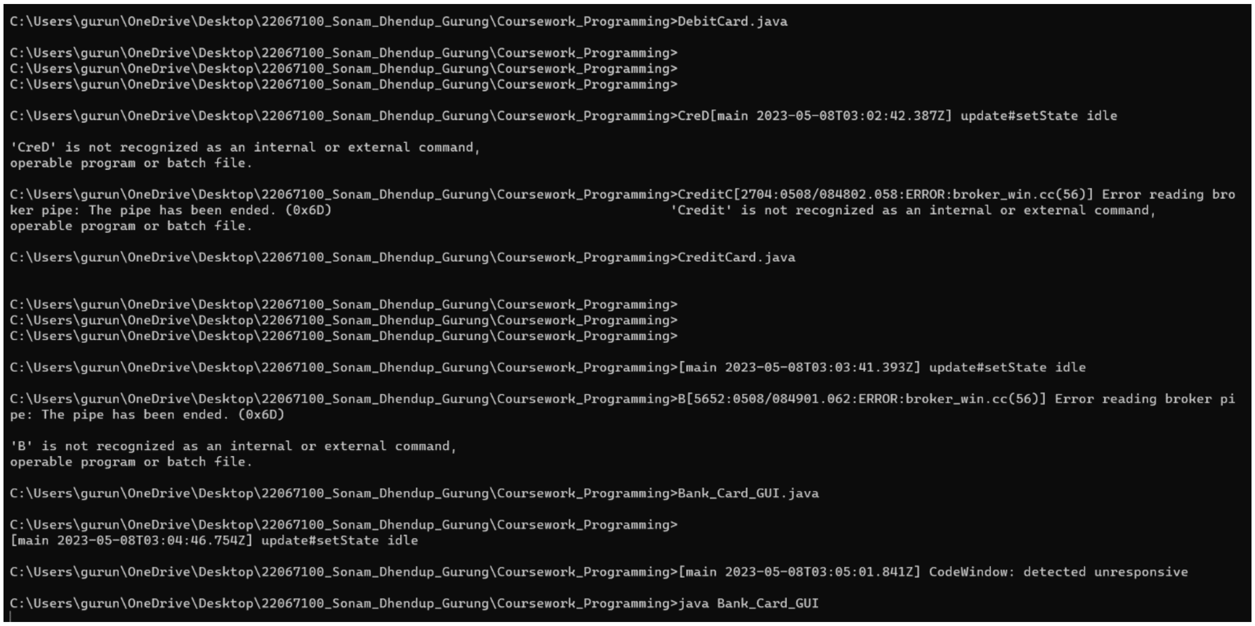
Method to use for compiling and running the program from Command Prompt.



```
{
public static void main(String [] args)
{
   Bank_Card_GUI zaza = new Bank_Card_GUI();
   zaza.m1D();
}
```

Figure 2: Method used in code for compiling and running

The program being compiled in Command Prompt:



```
C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>DebitCard.java

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>
C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>
C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>CreD[main 2023-05-08T03:02:42.387Z] update#setState idle

'CreD' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>CreditC[2704:0508/084802.058:ERROR:broker_win.cc(56)] Error reading bro
ker pipe: The pipe has been ended. (0x6D)                                                 'Credit' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>CreditCard.java


C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>
C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>
C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>[main 2023-05-08T03:03:41.393Z] update#setState idle

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>B[5652:0508/084901.062:ERROR:broker_win.cc(56)] Error reading broker pi
pe: The pipe has been ended. (0x6D)

'B' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>Bank_Card_GUI.java

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>
[main 2023-05-08T03:04:46.754Z] update#setState idle

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>[main 2023-05-08T03:05:01.841Z] CodeWindow: detected unresponsive

C:\Users\gurun\OneDrive\Desktop\22067100_Sonam_Dhendup_Gurung\Coursework_Programming>java Bank_Card_GUI
```

Figure 3: Command Prompt

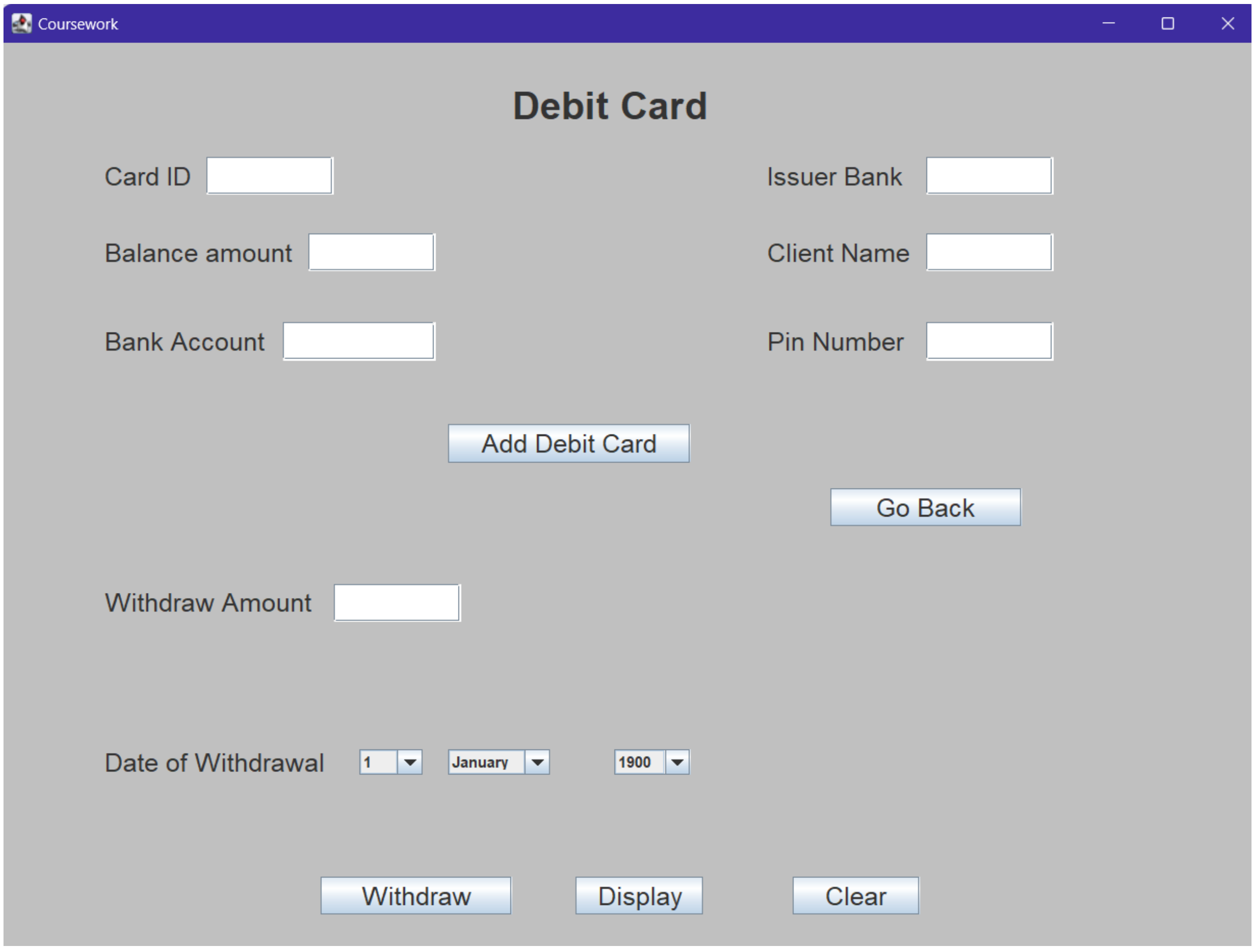


Figure 4 : GUI

GUI opened from the code executed in command prompt.

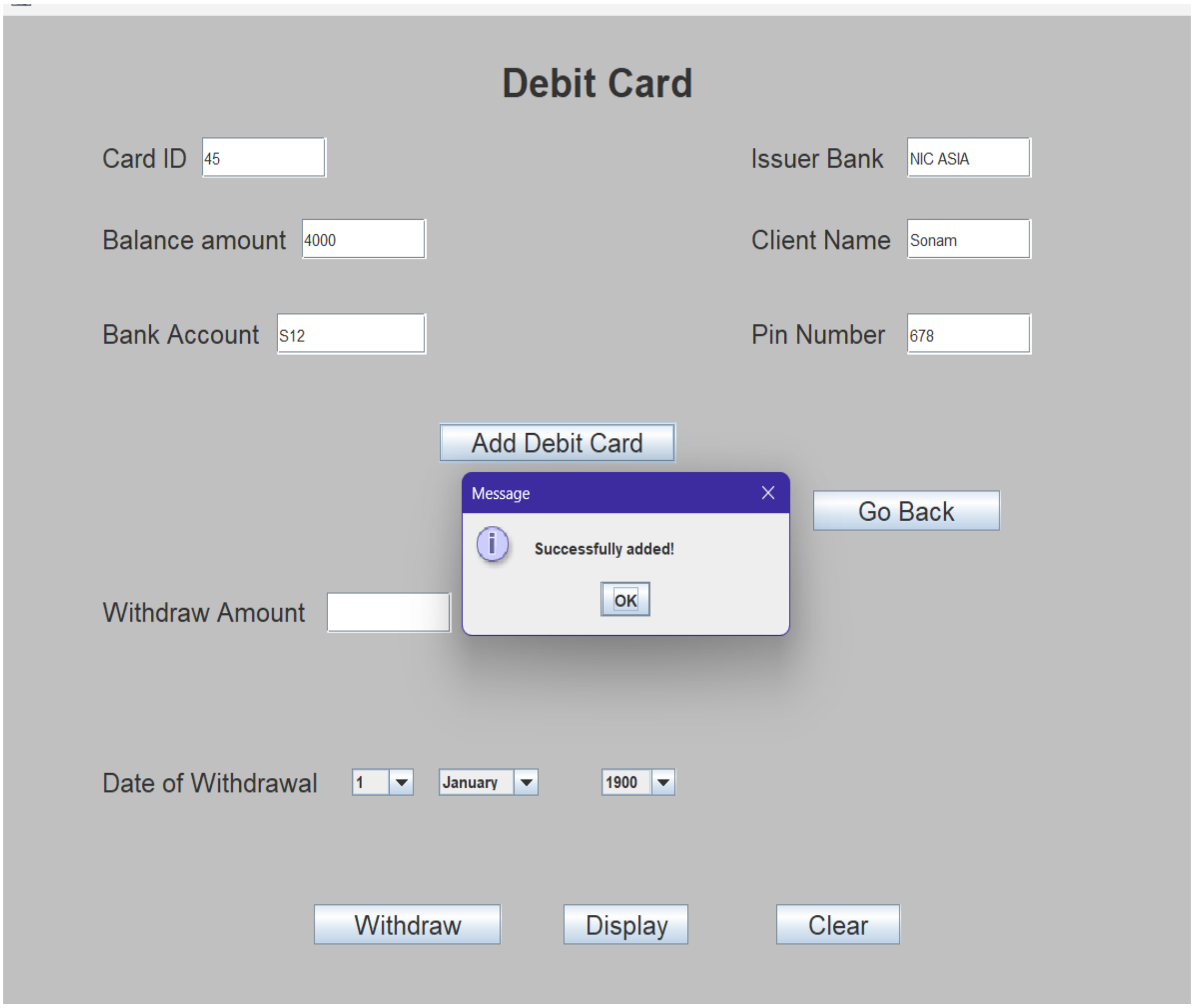
Test 2:

Table 2 : Inspect all the buttons

Test Number	2
Objective:	To inspect the buttons, Add Debit Card, Add Credit Card, Withdraw Amount from Debit Card, Set Credit Limit and Remove Credit limit.
Action:	To Call all the buttons with Evidence.
Expected Result:	The buttons functionalities should work and respond with dialogue boxes, the information of Debit card and credit card should be kept in its respective places. The amount should be withdrawn and credit limit should be set and the credit limit should also be removed.
Actual Result:	The buttons functionalities worked and responded with the dialogue boxes. The information of Debit card and credit card was kept in respective places. The amount was withdrawn, and credit limit was set and the credit limit was also removed.
Conclusion	The Test is successful.

Evidence

Add Debit Card



Debit Card

Card ID: 45

Issuer Bank: NIC ASIA

Balance amount: 4000

Client Name: Sonam

Bank Account: S12

Pin Number: 678

Message

Successfully added!

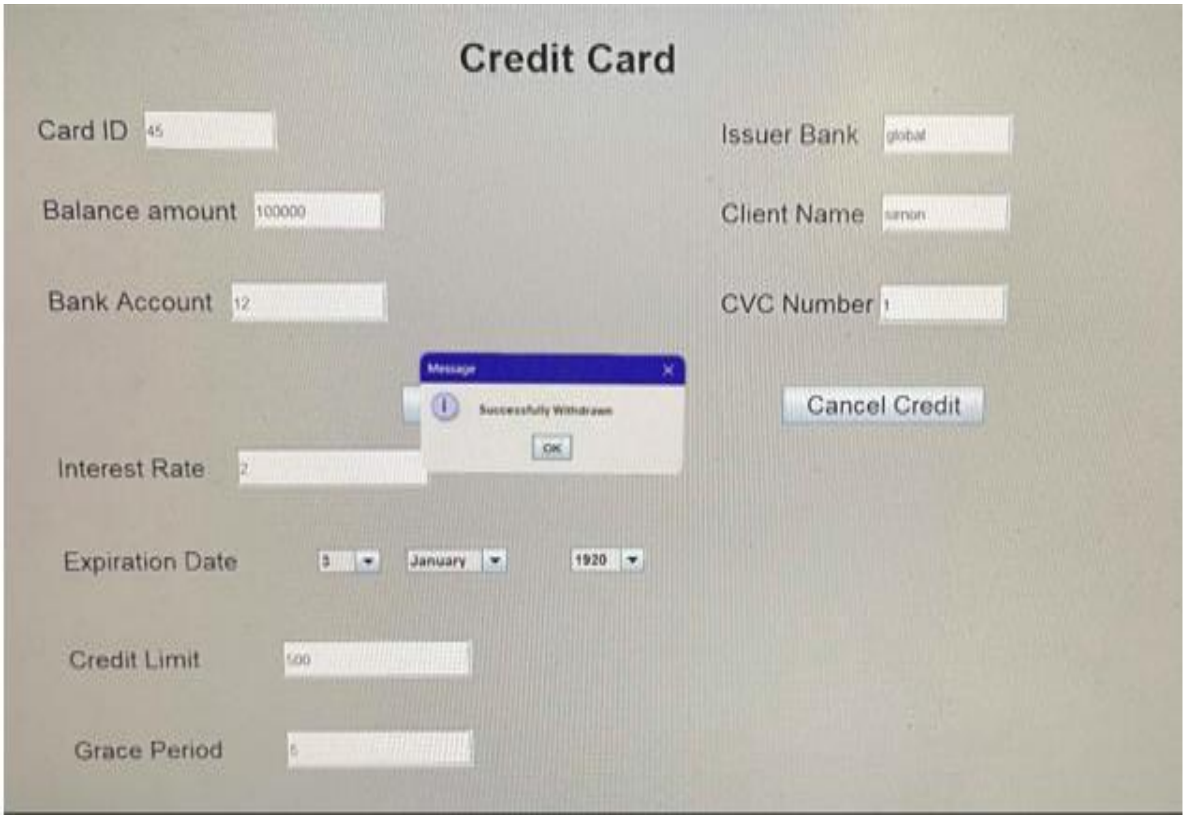
Withdraw Amount:

Date of Withdrawal: 1 January 1900

Figure 5 : Debit Card Added

In the above the values of card id, balance amount, Issuer bank, client name, bank account and Pin number is successfully added.

Add Credit Card



Credit Card

Card ID 45

Issuer Bank global

Balance amount 100000

Client Name sanon

Bank Account 12

CVC Number 1

Message

Successfully Withdrawn

Interest Rate 2

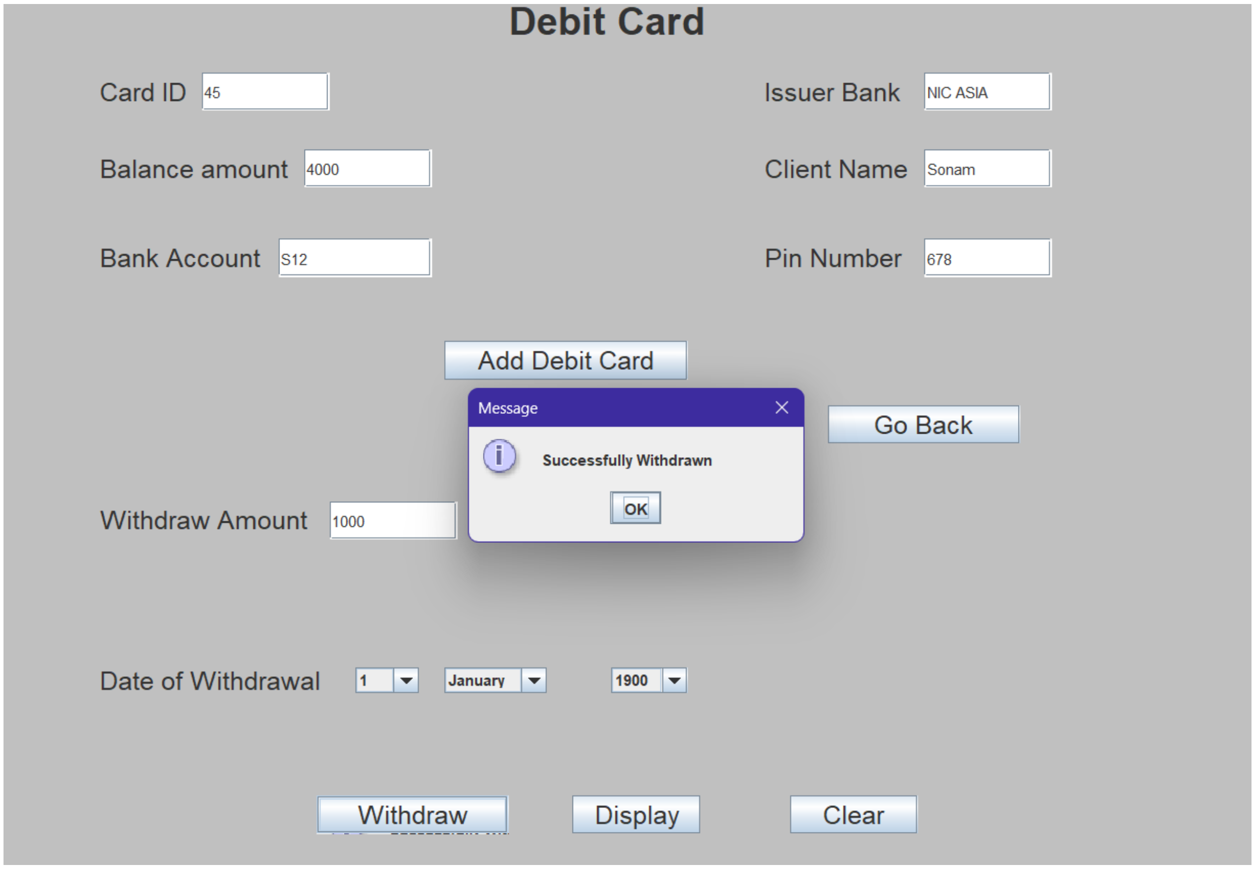
Expiration Date 3 January 1920

Credit Limit 500

Grace Period 5

In the above the values of card id, balance amount, Issuer bank, client name, bank account and CVC number, int rate , exp date ,credit limit is successfully added.

Withdraw from Debit Card



Debit Card

Card ID: 45

Issuer Bank: NIC ASIA

Balance amount: 4000

Client Name: Sonam

Bank Account: S12

Pin Number: 678

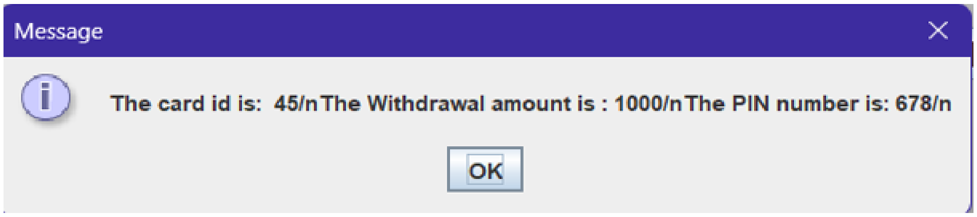
Withdraw Amount: 1000

Date of Withdrawal: 1 January 1900

Message

Successfully Withdrawn

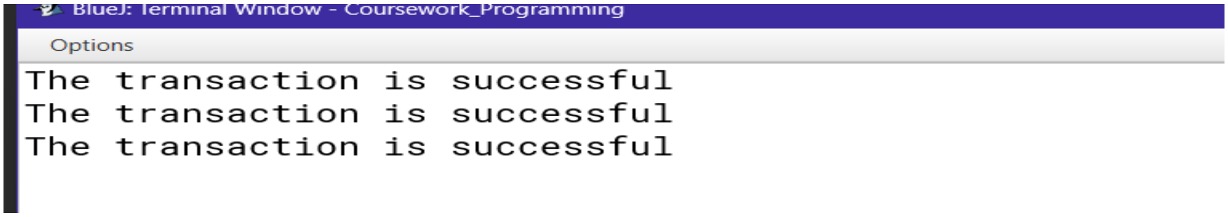
Figure 6: Withdraw from Debit Card



Message

The card id is: 45/nThe Withdrawal amount is : 1000/nThe PIN number is: 678/n

Figure 7: Details of the withdrawn amount

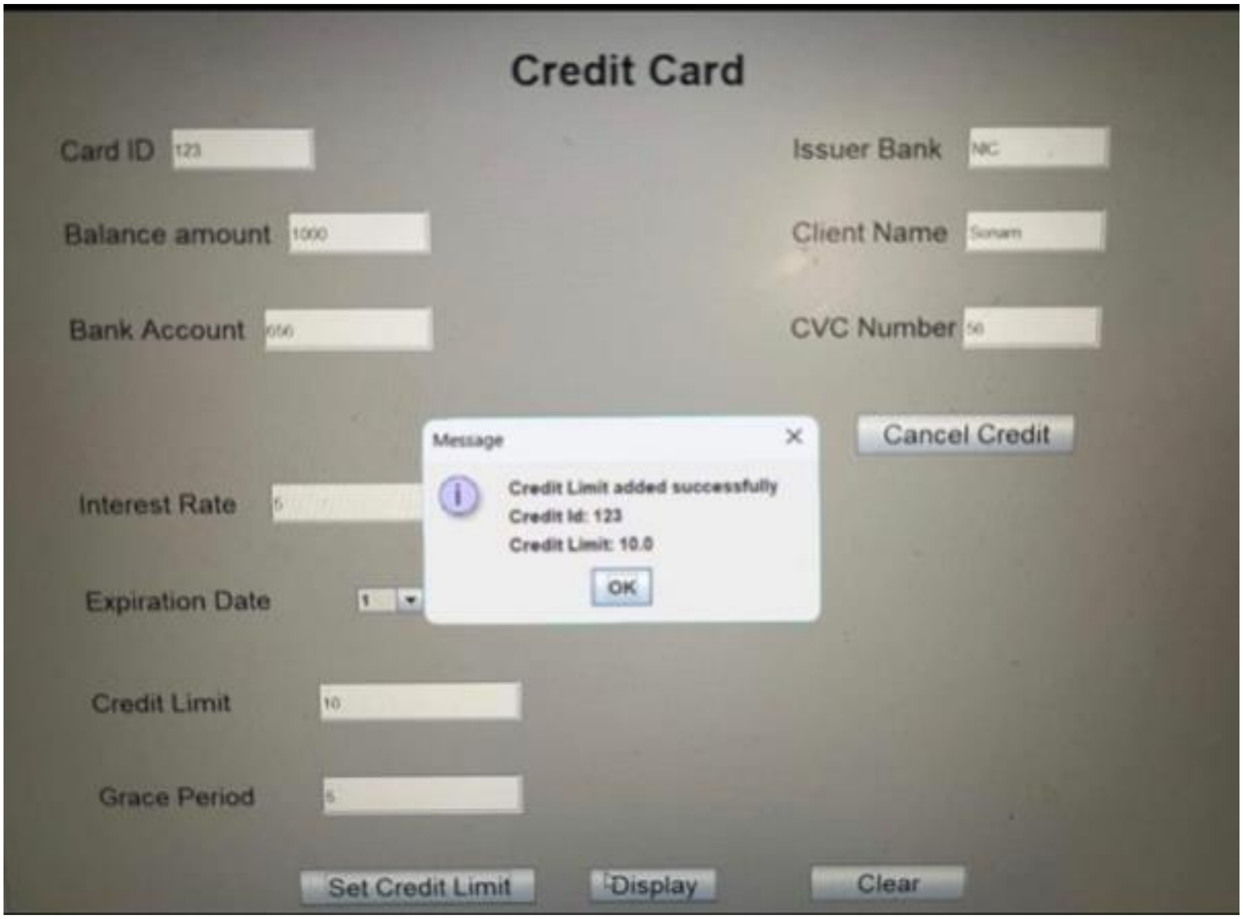


BlueJ: Terminal Window - Coursework_Programming

```
The transaction is successful
The transaction is successful
The transaction is successful
```

Figure 8: Transaction Complete

Set Credit Limit



Credit Card

Card ID 123

Issuer Bank NIC

Balance amount 1000

Client Name Sonam

Bank Account 656

CVC Number 56

Interest Rate 5

Expiration Date 1

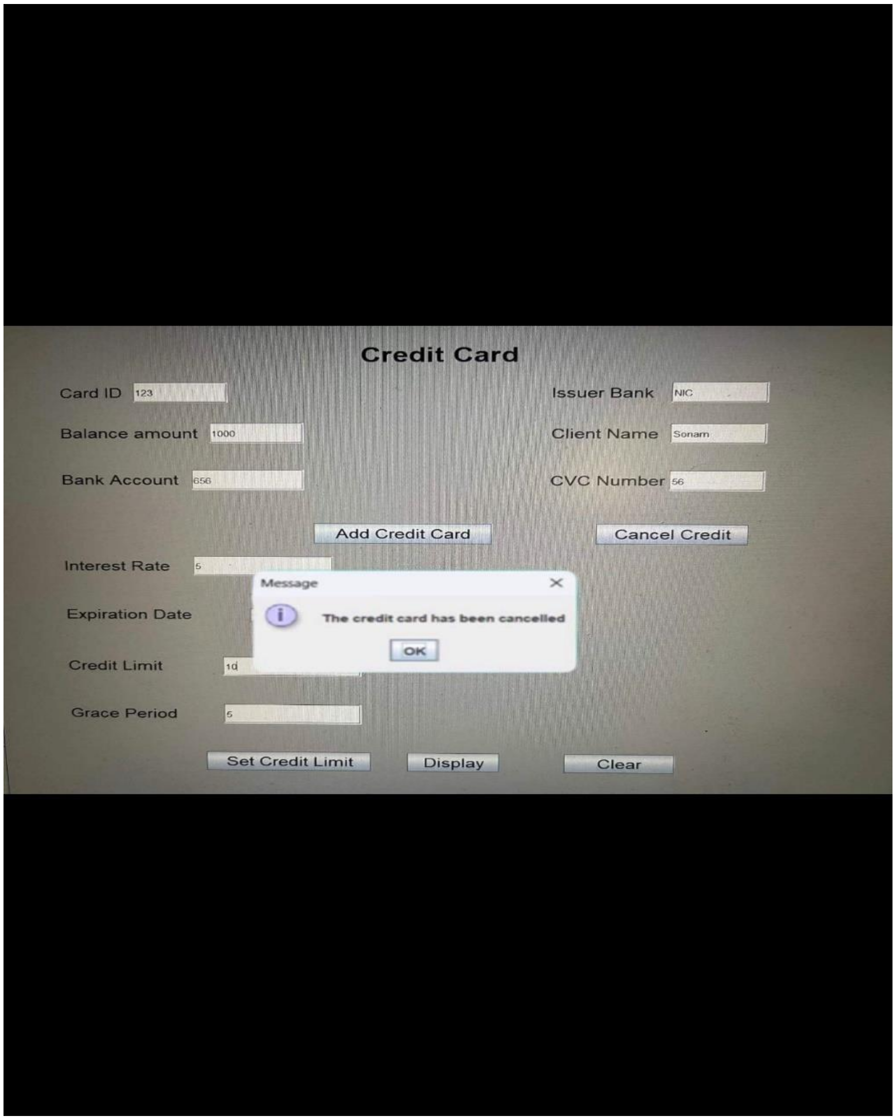
Credit Limit 10

Grace Period 5

Message

Credit Limit added successfully
Credit Id: 123
Credit Limit: 10.0

Remove Credit Card:



Credit Card

Card ID	123	Issuer Bank	NIC
Balance amount	1000	Client Name	Sonam
Bank Account	656	CVC Number	56
Interest Rate	5		
Expiration Date			
Credit Limit	1d		
Grace Period	5		

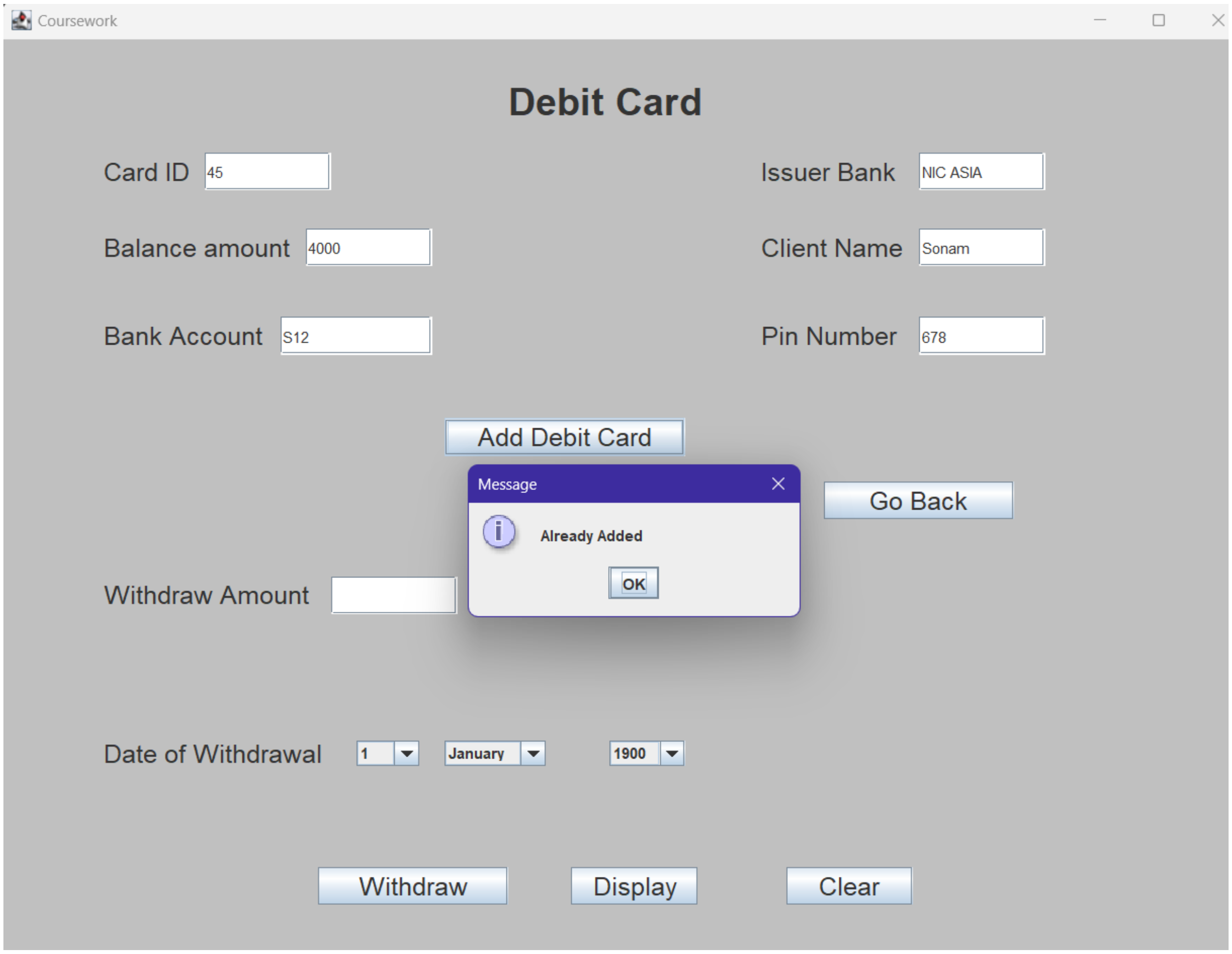
Message

The credit card has been cancelled

Test 3

1.Debit Card errors

The situation here is that the add button is clicked double which gives the dialog "Already Added."



Coursework

Debit Card

Card ID: 45

Issuer Bank: NIC ASIA

Balance amount: 4000

Client Name: Sonam

Bank Account: S12

Pin Number: 678

Withdraw Amount:

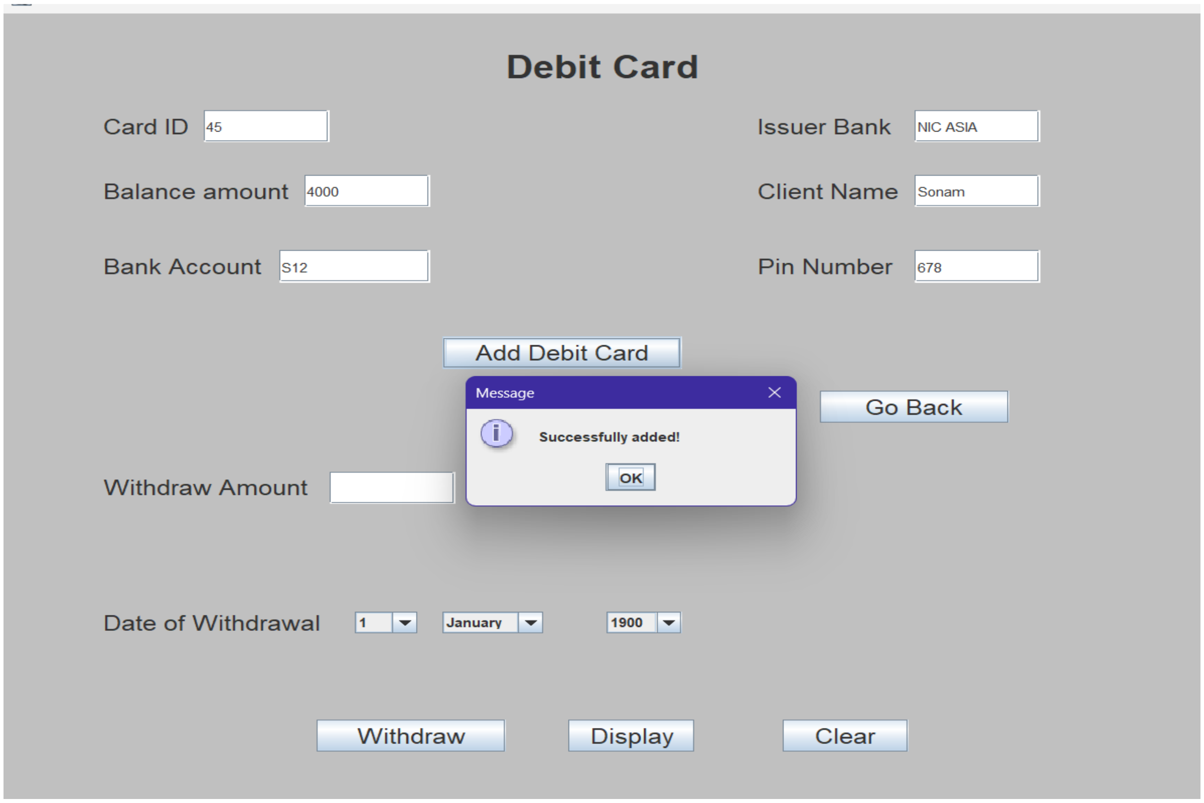
Date of Withdrawal: 1 January 1900

Message

Already Added

Figure 9: Debit Card Error

2. The Card Id doesn’t match error



Debit Card

Card ID: 45

Issuer Bank: NIC ASIA

Balance amount: 4000

Client Name: Sonam

Bank Account: S12

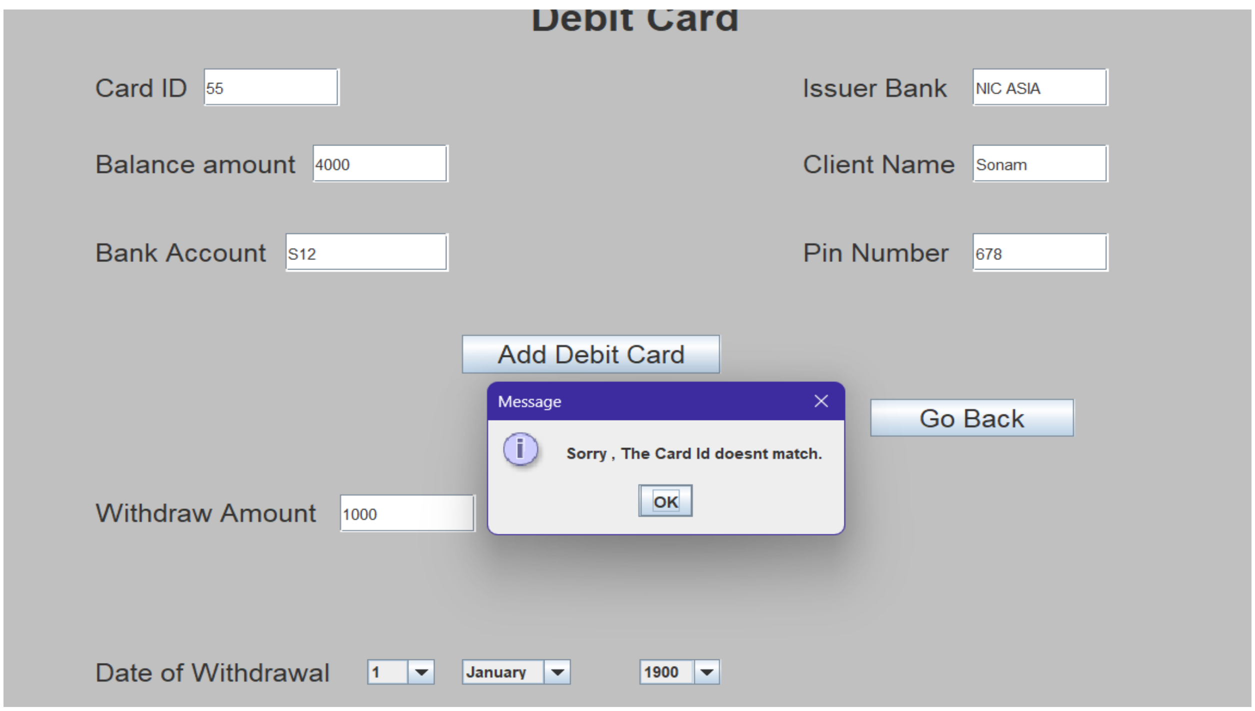
Pin Number: 678

Message

Successfully added!

Withdraw Amount:

Date of Withdrawal: 1 January 1900



Debit Card

Card ID: 55

Issuer Bank: NIC ASIA

Balance amount: 4000

Client Name: Sonam

Bank Account: S12

Pin Number: 678

Message

Sorry , The Card Id doesnt match.

Withdraw Amount: 1000

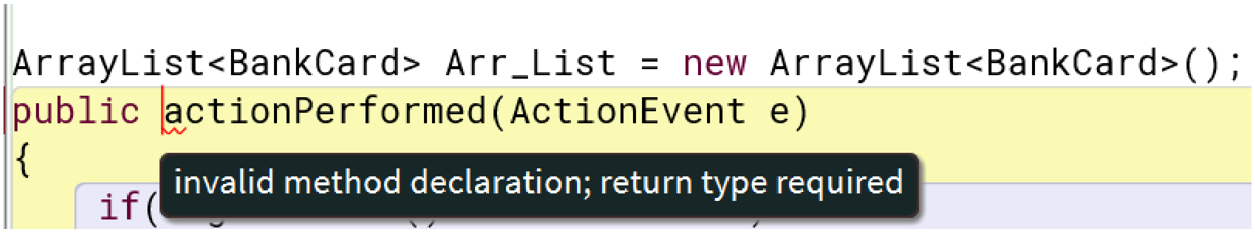
Date of Withdrawal: 1 January 1900

Figure 10: Card Id not matched

Error 1: Syntax Error:

1.Error Detected

The error was detected during declaration of the method. The error took place due to syntax error where the data type “integer” was missing.



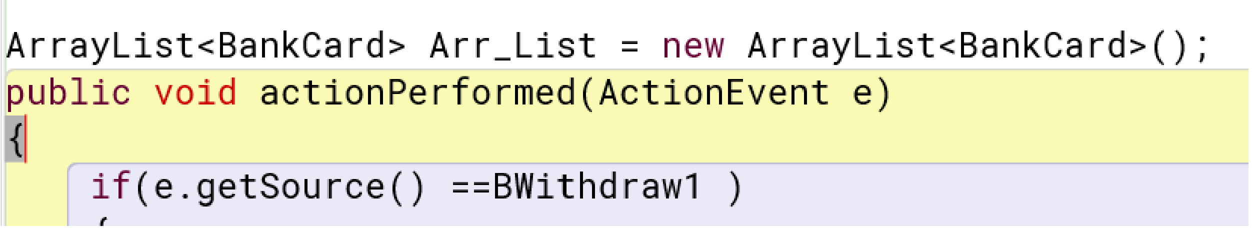
```
ArrayList<BankCard> Arr_List = new ArrayList<BankCard>();
public actionPerformed(ActionEvent e)
{
    if(
```

invalid method declaration; return type required

Figure 11: Syntax Error Detection

2.Error Correction:

The error was corrected by placing the missing method declaration “void.”



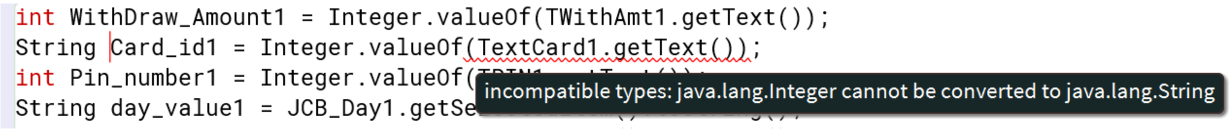
```
ArrayList<BankCard> Arr_List = new ArrayList<BankCard>();
public void actionPerformed(ActionEvent e)
{
    if(e.getSource() ==BWithdraw1 )
```

Figure 12: Syntax Error Correction

Error 2: Semantic Error

1.Error Detected:

The error was detected due to error in placement of datatype.



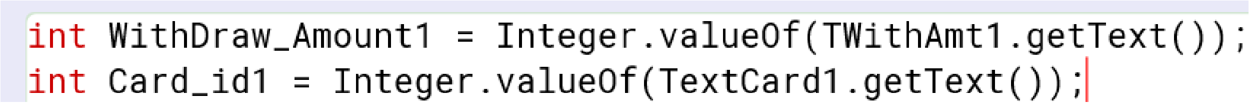
```
int WithDraw_Amount1 = Integer.valueOf(TWithAmt1.getText());
String Card_id1 = Integer.valueOf(TextCard1.getText());
int Pin_number1 = Integer.valueOf(
String day_value1 = JCB_Day1.getSe
```

incompatible types: java.lang.Integer cannot be converted to java.lang.String

Figure 13: Semantic Error Detection

2.Error Correction:

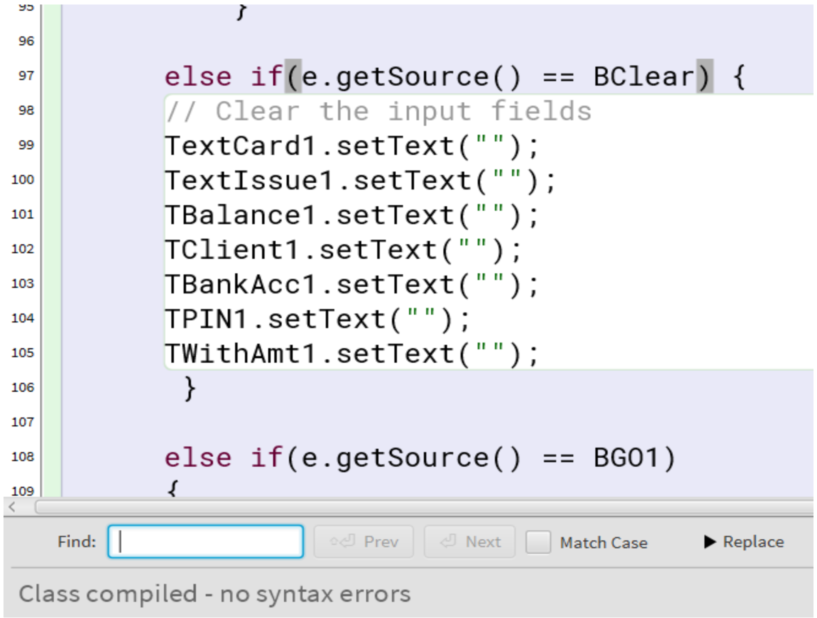
The error was corrected by replacing the datatype String with integer.



```
int WithDraw_Amount1 = Integer.valueOf(TWithAmt1.getText());
int Card_id1 = Integer.valueOf(TextCard1.getText());
```

Figure 14: Semantic Error Correction

Error 3: Logical Error



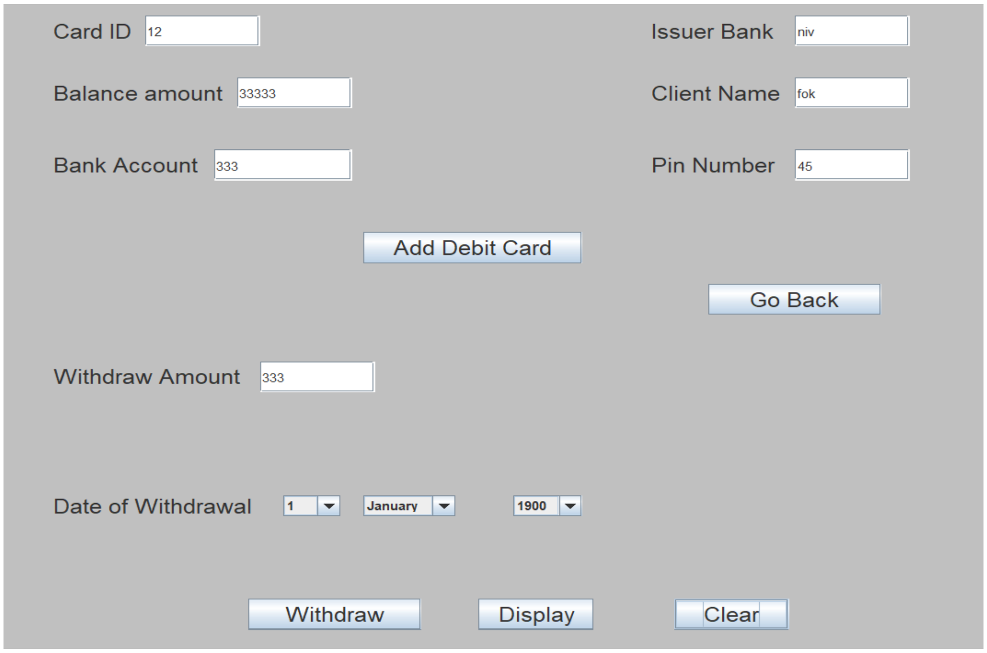
```
            }

        else if(e.getSource() == BClear) {
        // Clear the input fields
        TextCard1.setText("");
        TextIssue1.setText("");
        TBalance1.setText("");
        TClient1.setText("");
        TBankAcc1.setText("");
        TPIN1.setText("");
        TWithAmt1.setText("");
         }

        else if(e.getSource() == BG01)
        {
```

Class compiled - no syntax errors

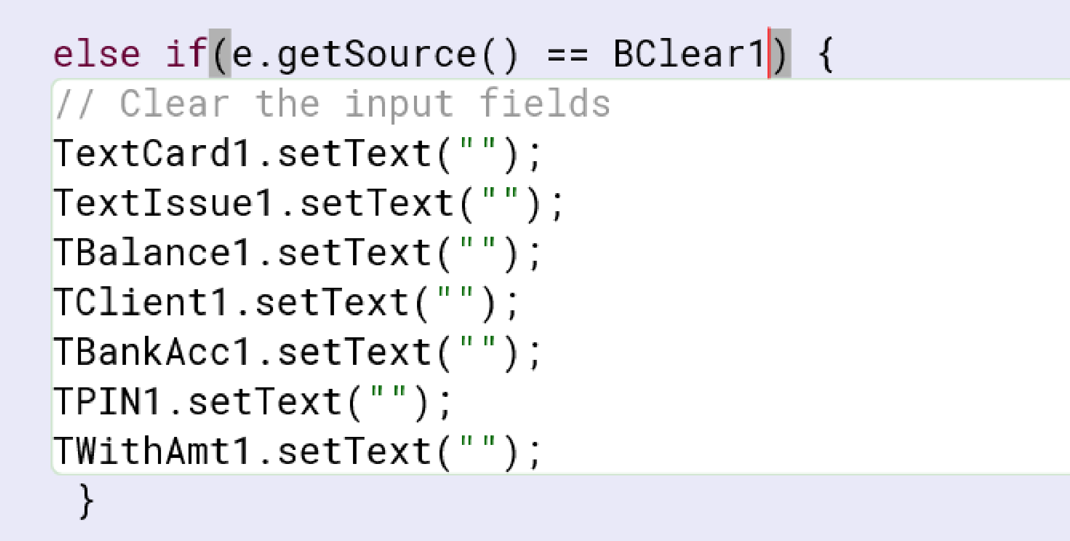
Here the code compiled.



Card ID	12	Issuer Bank	niv
Balance amount	33333	Client Name	fok
Bank Account	333	Pin Number	45
Withdraw Amount	333		
Date of Withdrawal	1 January 1900		

But the compiled code didn’t execute.

Error Detected:



```
else if(e.getSource() == BClear1) {
// Clear the input fields
TextCard1.setText("");
TextIssue1.setText("");
TBalance1.setText("");
TClient1.setText("");
TBankAcc1.setText("");
TPIN1.setText("");
TWithAmt1.setText("");
 }
```

Their was a simple logical error, the BClear1 button was to be kept instead of BClear.

Conclusion:

Evaluation of my work:

Comprehensively, it was a good experience on doing the coursework, I got the opportunity to learn about how Blue J works and how much helpful it is towards programming and all, I also got to learn about the syntax and other method used in the Blue J and implement it myself which was a new experience. Encountering and solving the errors was the most fun thing for me in this coursework. I could not solve a lot of problems at the beginning but was later I could due to the help of my tutor and colleagues and through research. I am very thankful to the university for giving me a chance to experience and learn new things.

I learned:

During and after the completion of the coursework, I was able to understand the mechanics of Blue J and Object-oriented languages and about java packages more deeply. At first, I didn't knew the use of methods and classes and parameters and GUI but with the right research and guidance, I am now able to succeed in completing the coursework and problem free from all the questions and problems. I am now able to create a Blue J language consisting of objects, parameters, constructers, Gui and other different methods.

Difficulties encountered:

During the coursework, I encountered many problems and difficulties, some taking me upto many hours to solve. Some of the mistakes were simple syntax errors while others were semantic errors due to wrong placements of data types and logical errors. Using Ms word was also a trouble since we had to adjust it to the university guidelines.

How you overcame these difficulties:

Most of the difficulties I faced was solved through research and consultation of my tutors and seniors and colleagues. I learned a lot especially from my seniors as they were quite helpful throughout the whole process. I am now clear of most of my doubts and finally now able to complete my coursework.

References

References

Ali, S. W., 2018. *Blue J Programming.* [Online]
Available at:
https://www.researchgate.net/publication/329140153_Blue_J_Programming
[Accessed 1 May 2023].

Anon., n.d. *Microsoft.* [Online]
Available at: https://ualr.edu/itservices/applications/v/microsoft-word/#:~:text=Microsoft%20Word%20is%20a%20word,both%20simple%20and%20complex%20documents.
[Accessed 1 May 2023].

Bibliography

Ali, S. W., 2018. *Blue J Programming.* [Online]
Available at:
https://www.researchgate.net/publication/329140153_Blue_J_Programming
[Accessed 1 May 2023].

Anon., n.d. *Microsoft.* [Online]
Available at: https://ualr.edu/itservices/applications/v/microsoft-word/#:~:text=Microsoft%20Word%20is%20a%20word,both%20simple%20and%20complex%20documents.
[Accessed 1 May 2023].